AT LAST!!

Cursor's New Years gift to you is getting this issue in your mailbox prior
to the 4th of July. Bally has invited the Cursor staff to the "International
Winter Consumer Electronics Show" being held in Las Vegas in January. Bally
is planning a private show in their Hilton suite. We will tell you all about
it in Issue #2.

We have included three programs in this issue, two of which are totally the
design (for better or worse) of Fred Cornett (Plastic Puzzle & Life Synthesis
Model), the other program is stolen from another publication, and has been
rewritten for the Bally (for the purpose of better understanding conversion
from one BASIC to another): "Electric Bill Analysis" is a translated version
of "POWER" by Karen S. Wolfe (written in North Star BASIC) in the October
issue of BYTE. I have had many, many requests for a tutorial regarding the
translation of printed programs written in other versions of BASIC.

For those of you desiring the ability to translate other BASIC programs to
Bally Tiny BASIC, I can only offer 2 bits of advice:

1. You must fully understand all Tiny BASIC statements before attempting
   any translation.
2. You must fully understand the BASIC statement you wish to translate.

Since most of us don't have the ability to work interchangeably with other
BASIC versions, I would, without reservation, recommend the purchase of
"The Basic Handbook" by David A. Lien. This book explains the purpose of
most statements utilized by all major versions of BASIC.

ELECTRIC BILL

This program utilizes the "internal calculator", which renders an answer
with 16 digit accuracy in the following format: NNNNNNNN.NNNNNNNN

This routine uses 18 string variables @(A), to load a number. The instruct-
ion is: $F@(A),@(B),@(C) Where F=Function (÷x-+)
  A=Beginning location of number to be manipulated
  B=Beginning location of number doing the manipulating
  C=Beginning location of Answer
I have bypassed the proper utilization of the internal calculator routine by
formatting, instead of leading zero suppression, etc. I did not use overflow
or sign indicators as they would not apply. Also, I did not make any attempt
to save bytes as I felt it was unnecessary. Enough space is left for consid-
erable expansion for those so desirous.

When prompted for $ Amount of electric bill, please input the Total of all
$ amounts, tax, fees, etc.

NOTE: You may have noticed that the Tiny BASIC statements used in this
       newsletter differ in format from the statements you are used to
       using. Do not alter the statements to conform to what you think
       the statement should look like; such as adding final quotes, etc.,
       the statements herein use a shorthand Tiny Basic which will work!

ELECTRIC BILL ANALYSIS
PROGRAM EXPLANATION

   1-163 Input Info
     18Ø Subtract beginning reading at @(18) from ending reading at @(Ø)
          and place answer (KW Hours) at @(54)
          Divide amount of bill at @(36) by KW Hours at @(54) and place
          answer (Price per KW Hour) at @(72)
     26Ø Clear registers at: @(Ø), @(18), @(36), @(54)
     27Ø Clear registers at: @(9Ø), @(1Ø8), @(126), @(144)
 35Ø-38Ø Input info
     39Ø Multiply wattage at @(Ø) by Hours of Appliance Use at @(126) and
          Place answer (Watt/Hours) at @(36)
          Clear @(Ø), @(18), and forces "1ØØØ" at @(Ø)
     395 Divide Watt/Hours at @(36) by "1ØØØ" at @(Ø) and place answer
          (Kilowatt/Hours) at @(18)
     4ØØ Multiply Kilowatt/Hours at @(18) by Price per KW/Hour at @(72) and
          place answer at @(54)
 41Ø-483 Print Answers

NOTE: All listings printed in Cursor utilize the same Special Character Set
       As your "ARCADE". In other words: What you see is what you type, i.e.,
       Bally Multiplication Sign=Lower Case X on listing (5x4).

┌──┐
│ AUTOMATIC RUN AFTER LOADING TAPE: When you are ready to save a program on │
│   tape; instead of typing   :PRINT; LIST   │
│   type: NT=1;::PRINT ;LIST ;PRINT ":RETURN ;TV=13;NT=3;RUN   │
│   Do not press go until tape is recording. After you press GO, the program │
│ will be saved on tape as before with the exception that upon completion of │
│ taping, it will print: :RETURN ;TV=13;NT=3;RUN on tape. When you input the│
│ program later, it will, upon completion of the load, run the program auto- │
│ matically even if you forgot to turn off the Tape Recorder.   │
└──┘

```
ELECTRIC BILL ANALYSIS  SZ=630

    1 NT=1
   10 CLEAR ;FOR A=ØTO 143;@(A)=Ø;NEXT A;BC=1Ø;FC=8
   20 PRINT "THIS PROGRAM ANALYZES    POWER USAGE AND COST
   40 FOR A=1TO 2ØØØ;NEXT A
  120 GOSUB 5ØØ;PRINT "INPUT ENDING METER READINGFROM ELECTRIC BILL (XXXXX)
  122 FOR E=12TO 85STEP -1;@(E)=KP;TV=@(E);NEXT E
  140 GOSUB 5ØØ;PRINT "INPUT BEGINNING METER";PRINT "READING FROM BILL (XXXXX)
  142 FOR B=3ØTO 26STEP -1;@(B)=KP;TV=@(B);NEXT B
  160 GOSUB 5ØØ;PRINT "INPUT $ AMOUNT OF BILL    (XXX.XX)
  162 FOR T=46TO 42STEP -1;IF T=43PRINT ".",
  163 @(T)=KP;TV=@(T);NEXT T
  180 $-@(Ø),@(18),@(54);$÷@(36),@(54),@(72)
  260 FOR A=71TO ØSTEP -1;@(A)=Ø;NEXT A
  270 FOR A=149TO 9ØSTEP -1;@(A)=Ø;NEXT A;CLEAR
  350 GOSUB 5ØØ;PRINT "SELECT AN APPLIANCE AND   INPUT WATTAGE (XXX)
  360 PRINT ;FOR W=1ØTO 8STEP -1;@(W)=KP;TV=@(W);NEXT W
  370 GOSUB 5ØØ;PRINT "INPUT THE ESTIMATED TOTAL HOURS OF APPLIANCE USE
  375 PRINT "DURING BILLING PERIOD    (XXXX)
  380 FOR H=137TO 134STEP -1;@(H)=KP;TV=@(H);NEXT H
  390 $x@(Ø),@(126),@(36);FOR A=35TO ØSTEP -1;@(A)=Ø;NEXT A;@(11)=49
  395 $÷@(36),@(Ø),@(18)
  4ØØ $x@(18),@(72),@(54)
  410 CLEAR
  420 PRINT "YOU PAY PER KWHOUR→$Ø.",
  422 FOR C=79TO 76STEP -1;TV=@(C);NEXT C;PRINT
  430 PRINT
  440 PRINT "APPLIANCE USED ",;FOR K=29TO 24STEP -1;IF K=25PRINT ".",
  445 TV=@(K);NEXT K;PRINT ;PRINT "KW HOURS OF POWER
  450 PRINT
  460 PRINT "BILLING PERIOD COST OF    THAT POWER USEAGE=$",
  465 FOR D=64TO 6ØSTEP -1;IF D=61PRINT ".",
  466 TV=@(D);NEXT D;PRINT
  470 PRINT
  480 PRINT "COST PER EST. HOUR'S USE =$Ø.",
  482 $÷@(54),@(126),@(1Ø8)
  483 FOR Z=115TO 112STEP -1;TV=@(Z);NEXT Z
  490 PRINT "  PRESS ANY NUMBER",
  495 A=KP;GOTO 260
  5ØØ CLEAR ;PRINT "PLEASE PRECEDE WITH ZEROES";PRINT ;RETURN
```

BYTESAVING HINTS:

By now, some of you have input the preceding program and noticed that the number of bytes remaining (obtained by PRINT SZ) on your machine is somewhat less than we have stated.

A good habit to use whenever you desire to print an instruction or comment on screen, is to make as much use of the command words as possible (RUN, CLEAR, INPUT, ETC.). These command words occupy 1 byte total versus 1 byte per letter if input letter by letter.

PLASTIC PUZZLE

This program was challange for me, as I needed one more program to show the versatility of the PX function. I would imagine the game is just as boring as the 50¢ plastic version. The non-programmer will look at the display and say "Big deal!". The programmer will ask how the letter being moved knows where the blank space is. So, I have printed this program to offer a sharp little PX Routine and not for the abject thrill of playing the game.

For those of you that are unfamiliar with this game, the object is to move the letters around until they are in alphabetical order. One project you programmers might consider, is writing an end to the game when the letters are all in proper order.

PLASTIC PUZZLE PROGRAM EXPLANATION:

```
        2 Set Colors & divide screen
  10-  30 Creat game board
  50-  70 Selects random letters(A-O no duplicates)& places them on-screen.
 100- 140 Input letter to be moved
 150- 225 Go to location of letter to be moved and see if an open square
          is adjacent to it (not diagonal)
      227 If open square exists inside game board adjacent to letter to be
          moved-clear square of letter to be moved
      229 Same as 227 except creates moved letter in new square
      230 If no open square exists adjacent to letter to be moved (inside
          game board) Illegal Move-Restart sequence
     1001 Clears left side of screen
1200-1210 Positive Pixel Search parameters
```

```
PLASTIC PUZZLE  By: F. Cornett  SZ=982
   1 CLEAR
   2 NT=3;&(Ø)=89;&(1)=89;&(2)=163;&(3)=163;BC=89;FC=11;&(9)=28
   5 C=1ØØØ;D=C+1
  1Ø CLEAR ;BOX 54,2Ø,41,41,1;FOR A=1TO 33;Ø(A)=Ø;NEXT A
  2Ø FOR Y=35TO 55STEP -1Ø;FOR X=39TO 69STEP 1Ø
  3Ø BOX X,Y,9,9,2;CX=X+1;CY=Y
  5Ø A=RND (16);IF Ø(A)GOTO 5Ø
  52 IF A=16TV=32;Ø(A)=X;Ø(A+16)=Y;GOTO 7Ø
  6Ø TV=A+64;Ø(A)=X;Ø(A+16)=Y
  7Ø NEXT X;NEXT Y
 1ØØ GOSUB C;GOSUB D;PRINT "INPUT LETTER";PRINT "YOU WISH TO ";PRINT "MOVE USING";
     PRINT "KN(1) & TR(1)
 11Ø CY=8;CX=-71;NT=Ø
 12Ø L=KN(1)÷17+72;TV=L
 13Ø IF TR(1)GOTO 15Ø
 14Ø GOTO 11Ø
 15Ø NT=1;X=Ø(L-64);Y=Ø(L-48)
 2ØØ G=Ø;H=Ø;S=Y;E=Ø;F=X+1Ø;GOSUB 12ØØ
 21Ø S=Y-1Ø;F=X;GOSUB 12ØØ
 22Ø S=Y;F=X-1Ø;GOSUB 12ØØ
 225 S=Y+1Ø;F=X;GOSUB 12ØØ
 227 IF E<4IF GIF HØ(16)=X;Ø(32)=Y;CX=X;CY=Y;TV=32
 229 IF E<4IF GIF HØ(L-64)=G;Ø(L-48)=H;CX=G;CY=H;TV=L;GOTO 1ØØ
 23Ø GOSUB D;GOSUB C;PRINT "ILLEGAL";PRINT "MOVE";PRINT ;PRINT ;GOTO 1ØØ
1ØØØ CX=-77;CY=4Ø;RETURN
1ØØ1 BOX -23,Ø,114,88,2;RETURN
12ØØ IF PX(F-2,S+3)E=E+1;RETURN
12Ø5 IF PX(F+2,S+3)E=E+1;RETURN
12Ø6 IF PX(F+2,S-3)E=E+1;RETURN
12Ø7 IF PX(F,S-3)E=E+1;RETURN
12Ø8 IF PX(F-2,S-3)E=E+1;RETURN
12Ø9 IF (F<39)+(F>7Ø)+(S>35)+(S<5)E=E+1;RETURN
121Ø G=F;H=S;RETURN
```

The following article has been printed here due to the patience and lar-
gess of Ed Mulholland, who lacking any formal education in this field,
beat his way through all the printed matter available to finally arrive
bearing a functional full sized ASCII Keyboard interfaced to the Arcade
(Why don't the "professionals" among us ever share information?). This
wearied warrior still awaited yet one more gruesome test of his patience;
that of instructing the Cursor Editor what the hell this all meant. Whew!!

We have it on good authority (Mrs. Cornett) that the Cursor Editor is a
functional idiot when it comes to putting things together. Therefore, we
very strongly suggest you run right out and buy "TV Typewriter Cookbook"
by Don Lancaster. This book contains rather complete instructions &
information on such things as - Baud-Rate Generators--Keyboard Encoders--
PROM--RAM--Serial Interface UARTs--Bus Organization--What to look for when
buying a keyboard--Radio Data Links--Modems--Printers, etc. If you aren't
an old hand at wiring up your own projects - buy Lancasters book before
you do anything else.

As Ed Mulholland points out - there are a number of ways of adding a key-
board, I wrote Jerry Tindle (who has also added a keyboard) over a month
ago and have not, as yet, received an answer. If you have a keyboard or
any other device, please drop the Cursor a line and let us know something
about it!

FULL SIZE ASCII KEYBOARD
By: Edmund Mulholland

The basic idea is to provide serial data from a keyboard through the
Cassette Interface at 150 baud and then read this data with a :INPUT.

I used an AY51013A UART board sold by Electronics Systems because I was
not sure what I might need to develop this system, and also because it
provides various methods of manipulating. For those who want to save
some money, my schematic is shown without this additional circuitry. The
keyboard encoder is an AY53600. I chose this encoder because it has a
latched output rather than a continuous stream of data. You release this
data by taking a negative strobe from Pin 16 of the encoder and applying
to Pin 23 of the UART. I used a .1mf capacitor for a longer key debounce
time, and used the clock from the interface to synchronize the UART. It
may be possible to use other methods that would work; but, being anxious
to use the keyboard, I left it this way. Also, a TPST switch is used to
isolate the data entries. The addition of the 2.2K and 22K resistors
eliminate the need for the switch, but it would not hurt to have a method
of isolating these signals.

I have not been able to get the inputs ↓\PAUSE GO+10 HALT. As you can
see, these characters are only a minor inconvenience and can be added to
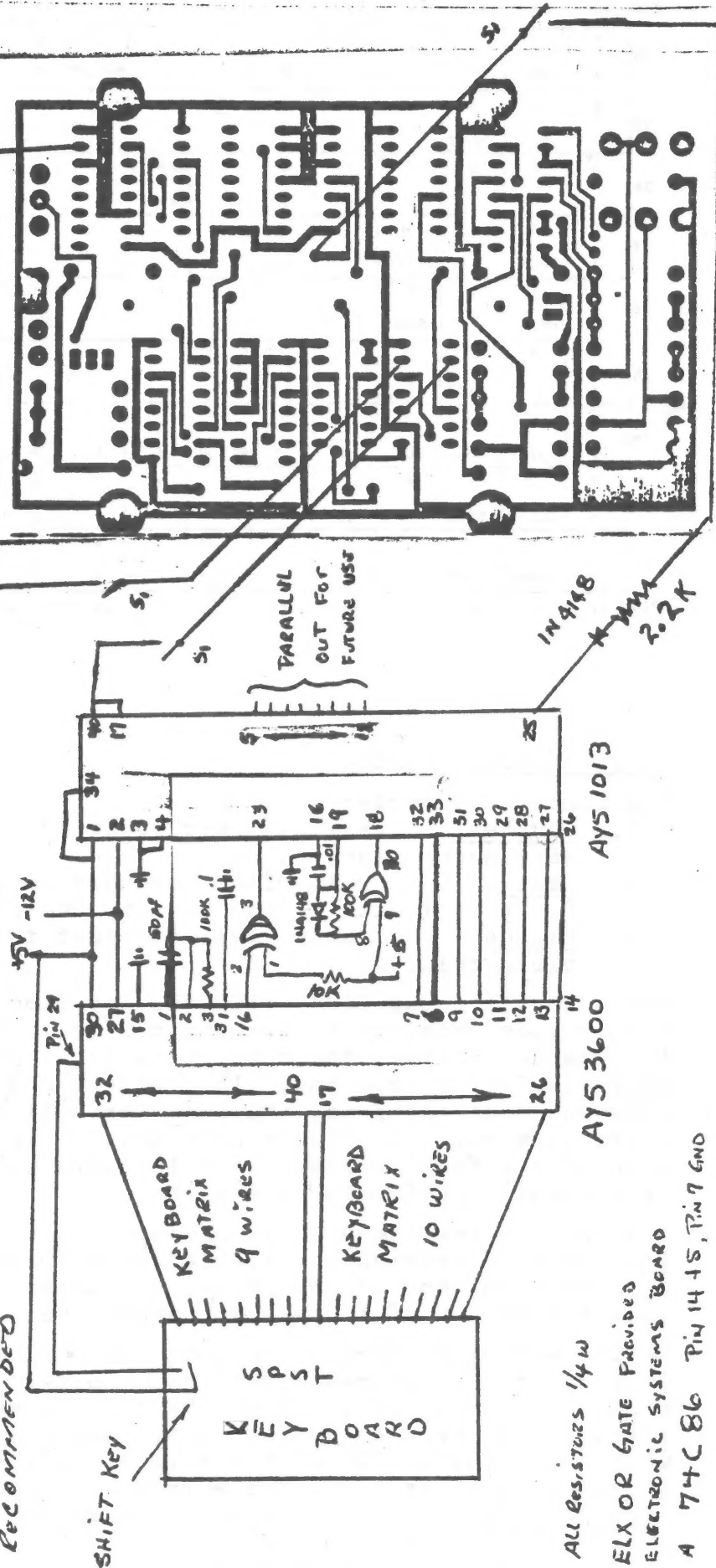the program later. Also, other characters are available as new graphic
symbols.

The cost of the keyboard for this project is approximately $50.00, not
including a power supply. I chose to use a separate power supply because
of not knowing how much power the Bally unit could provide, and also be-
cause I neek power for future additions. My choice was JAMECO, the JE200
and JE205. If this is your choice, pay close attention to the instructions
on winding the toriodal transformer. This addition increases the cost to
approximately $80.00. If the Electronics Systems board (with parts) is
used, the cost is approximately $110. By shopping the surplus stores and
maybe using a smaller power supply, you can cut your cost to approximately
$50.00 or less.

Only a few keys need to be remarked from a standard keyboard, and this is
left up to the user to decide. You can check the truth table and decide
what you want.

FULL SIZED ASCII KEYBOARD PARTS LIST

JAMECO ELECTRONICS	JE200 Regulated Power
1021 Howard Avenue	Supply Kit $14.95
San Carlos, CA 94070	JE205 Multi-Voltage
(415) 592-8097	Board Kit $12.95
ELECTRONICS SYSTEMS	UART & Baud Rate Generator
P.O. Box 21638	Part #: 101AW/parts
San Jose, CA 95151	$35.00

Switch S1 is optional
But Recommended

SHIFT Key

KEYBOARD MATRIX 9 wires

KEYBOARD MATRIX 10 wires

KEYBOARD SPST

PARALLEL OUT FOR FUTURE USE

AY5 1013

AY5 3600

22K

2.2K

1N4148

+5V -12V

ALL Resistors 1/4 w

ELX OR GATE Provided
CH Electronic Systems Board
is a 74C86 Pin 14 +5, Pin 7 GND

PIN	17	18	19	20	21	22	23	24	25	26	PIN
40	<	Q	A	Z		H	+	>	■	!	40
	1	BOX	→			H	+		RETURN	1	
39	■	W	S	X		%	⅂		↑	"	39
	2	TO	INPUT			%	IF		GOTO	2	
38	#	E	D	C	←	$	L	ERASE	&	[	38
	3			÷	←	$	L	ERASE	6	NEXT	
37	$	R	F	SPACE	(	GO			'	"	37
	4	FOR		SPACE		GO			7	"	
36	%	T	G	V				■	)	SPACE	36
	5	PRINT		RND				-	)	SPACE	
35	>	Y	H	B	*	>	+		*	!	35
	6		LIST	x times	:	>	;		*	!	
34	&	U	J	N	=	<	P	)	&	#	34
	7	STEP	RUN	GOTO	=	<	RETURN	Ø	&	#	
33	*	I	K	M	?	"		+	<	(	33
	8	CLEAR	NEXT	IF	/	'		■		(	
32	(	O	L	,	.	:	[	←	Ø	)	32
	9	GOSUB	LINE	,	.	;	⅂	-	Ø	9	
PIN	17	18	19	20	21	22	23	24	25	26	PIN

Keep in mind, Bally uses eight bits, with the MSB bit always a zero. This truth table works only when wired as shown; otherwise, you will get different results. The Control key is not used, but some users may want to try various ideas.

Anyone wanting additional May write:

Mr. Edmund Mulholland
Route 4, Box 424 H
N. Wilkesboro, NC 28659

You must include a stamped self-addressed envelope if you desire a reply!

If you decide to build this project, please let the Cursor know of your results and problems.

LIFE SYNTHESIS MODEL

The rules are Simple:
1. <u>Survival</u>. Every cell with 2 or 3 adjacent cells survives for the next generation.
2. <u>Death</u>. Each cell with 4 or more neighbors dies from overpopulation. Every cell with 1 neighbor or none dies from isolation.
3. <u>Births</u>. Each empty cell adjacent to exactly 3 live cells is a birth cell.

Using the grid supplied, plot the location of your starter colony. Best results are obtained by keeping the cells in one area. After inputting the starter colony, the computer will search the complete field, taking approximately 25 minutes. This will occur only during generation #1, on subsequent generations, the computer will only search those quadrants where life exists. Please note the "search locators" on the top and left side of the field border. The locators will show you where the computer is currently performing a search.

I know of a few areas where a minimal amount of memory could be saved, such as substituting a variable for GOTO (5Ø) or (61), but I would like very much to know if any of you can speed up the search process without sacrificing any of the search parameters.

NOTE: When inputting a program, pay strict attention to spacing, etc. In many of the programs printed in Cursor, there are not more than a couple of bytes left. If you have input spaces that are not shown in the listing, the program may not run. Also, please make sure you proofread what you input!! 90% of the time an individual thinks a program contains glitches, they are operator errors. EXAMPLE:
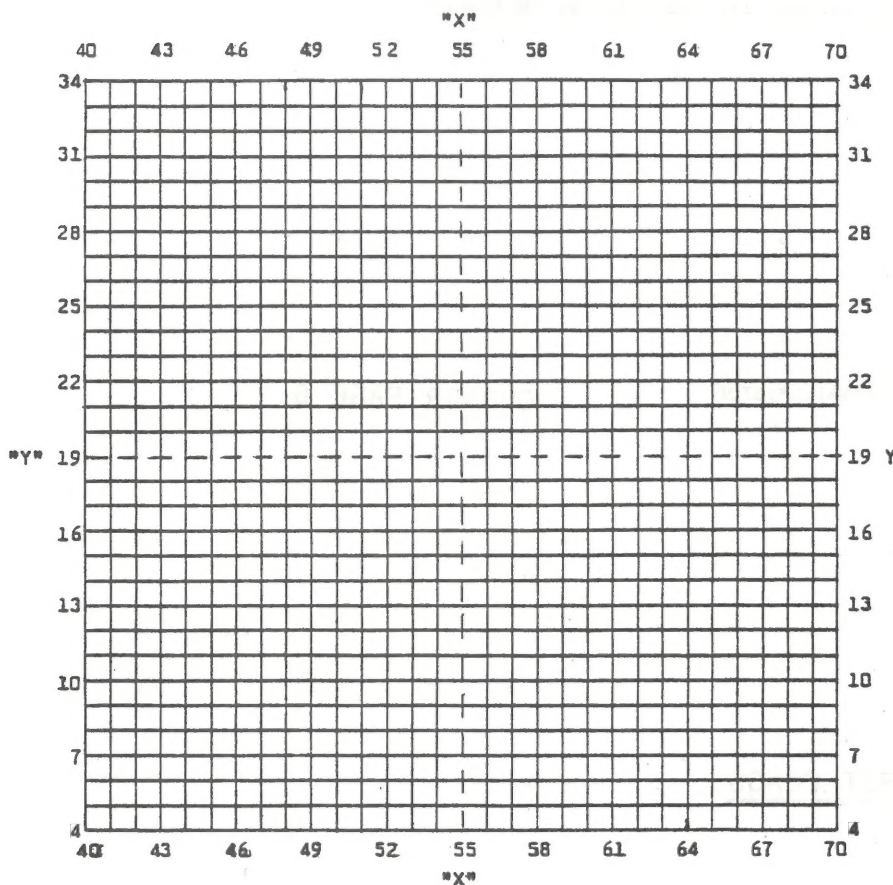
<u>100 IF A=BC=A</u>   Will not work. You cannot put two characters of the same type together!

<u>11Ø IF A=B@(1)=A</u>   Will work!

<u>120 IF A=BIF C=DGOTO 1Ø</u>   Will Work

LIFE SYNTHESIS MODEL PROGRAM EXPLANATION

```
  1-  15 Input locations of Starter Colony
      20 Put up field border
 21-  22 Put up grid marks
      30 Put up starter colony
 40-  49 Sets Horizontal Search Parameters (using Black pixel locators
                                            at 615-630)

 51-  60 Sets Vertical search parameters
 80-  81 Sets Colors and splits screen
100- 103 Puts up search locators
110- 120 Determines if Pixel is black or white-sends to
         appropriate subroutine
     130 Erases search locators if current search has ended
     500 If Pixel is white-Send to subroutine that checks status of
         adjacent pixel locations
     510 Creates new cell & updates "Birth" count- changes color to
         green and back
     600 If pixel is black-Sends to subroutine that checks status of
         adjacent pixel locations
     610 Erases cell and updates "Deaths" count-changes color to
         red and back
615- 630 Sets black pixel locators
1000-1080 Adjacent pixel search
    2000 Updates "Population" count
    2010 If Population=0  Terminates program
```


STARTER COLONY
LOCATOR GRID

SAMPLE STARTER COLONY

```
1-X=45  Y=31
2-X=45  Y=30
3-X=46  Y=31
4-X=46  Y=29
5-X=46  Y=28
6-X=47  Y=29
7-X=48  Y=31
8-X=48  Y=30
9-X=49  Y=31
```

LIFE SYNTHESIS MODEL   By: Fred Cornett   SZ=44

```
    1 &(9)=50;CLEAR ;PRINT "LIFE SYNTHESIS
    3 B=0;D=0;Z=1;E=36;F=75;I=0;H=39;Q=0;T=0;S=0;R=0;L=0;M=0;N=0;P=9;O=0
    5 PRINT ;PRINT "INPUT (X,Y)FOR 9 CELLS
   10 FOR A=1TO 9;PRINT #1,"CELL #",A;INPUT "X"@(A)
   15 INPUT "Y"@(A+10);NEXT A;W=118
   20 CLEAR ;BOX 56,19,48,48,1;BOX 56,19,46,46,2
   21 FOR G=36TO 76STEP 10;BOX G,41,1,1,1;BOX G,-4,1,1,1;NEXT G
   22 FOR U=0TO 40STEP 10;BOX 33,U,1,1,1;BOX 78,U,1,1,1;NEXT U
   25 CX=-77;CY=-16;PRINT #1,"POPULATION=",P
   30 FOR A=1TO 9;BOX @(A),@(A+10),1,1,1;NEXT A
   40 IF QIF T E=36;F=75;GOTO 50
   41 IF QIF T=0IF S E=36;F=66;GOTO 50
   42 IF QIF T=0IF S=0IF R E=36;F=56;GOTO 50
   43 IF QIF T=0IF S=0IF R=0E=36;F=46;GOTO 50
   44 IF Q=0IF RIF T E=46;F=75;GOTO 50
   45 IF Q=0IF RIF SIF T=0E=46;F=66;GOTO 50
   46 IF Q=0IF RIF S=0IF T=0E=46;F=56;GOTO 50
   47 IF Q=0IF R=0IF SIF T E=56;F=75;GOTO 50
   48 IF Q=0IF R=0IF S=0IF T E=66;F=75;GOTO 50
   49 IF Q=0IF R=0IF SIF T=0E=56;F=66
   50 Q=0;R=0;S=0;T=0
   51 IF LIF O H=39;I=0;GOTO 61
   52 IF LIF O=0IF N H=39;I=10;GOTO 61
   53 IF LIF O=0IF N=0IF M H=39;I=20;GOTO 61
   54 IF LIF O=0IF N=0IF M=0H=39;I=30;GOTO 61
   55 IF L=0IF MIF O H=30;I=0;GOTO 61
   56 IF L=0IF MIF NIF O=0H=30;I=10;GOTO 61
   57 IF L=0IF MIF N=0IF O=0H=30;I=20;GOTO 61
   58 IF L=0IF M=0IF NIF O H=20;I=0;GOTO 61
   59 IF L=0IF M=0IF N=0IF O H=10;I=0;GOTO 61
   60 IF L=0IF M=0IF NIF O=0H=20;I=10
   61 L=0;M=0;N=0;O=0
   80 &(0)=116;&(1)=116;&(2)=118;&(3)=118
   81 BC=116;FC=114;&(9)=27
   90 CY=40;PRINT "CURRENT
   99 CY=32;PRINT #1,"GENERATION #",Z
  100 FOR X=ETO F;FOR Y=HTO ISTEP -1
  103 BOX X,43,1,1,1;BOX 31,Y,1,1,1
  110 IF PX(X,Y)=0GOSUB 500
  120 IF PX(X,Y)GOSUB 600
  130 NEXT Y;BOX 31,19,1,48,2;NEXT X;BOX 55,43,48,1,2;Z=Z+1;GOTO 40
  500 C=0;GOSUB 1000
  510 IF C=3&(2)=147;&(3)=147;BOX X,Y,1,1,1;B=B+1;CY=24;PRINT #1,"BIRTHS=",B;
      C=0;P=P+1;&(2)=W;&(3)=W;GOSUB 2000;RETURN
  520 C=0;RETURN
  600 C=0;GOSUB 1000
  610 IF (C>3)+(C<2)&(2)=90;&(3)=90;BOX X,Y,1,1,2;D=D+1;CY=16;PRINT #1,"DEATHS=",D;
      C=0;P=P-1;&(2)=W;&(3)=W;GOSUB 2000;RETURN
  615 IF X<46Q=1
  616 IF X<56IF X>45R=1
  617 IF X<66IF X>55S=1
  618 IF X>65T=1
  619 IF Y>29L=1
  620 IF Y>19IF Y<31M=1
  621 IF Y>9IF Y<21N=1
  622 IF Y<110=1
  630 C=0;RETURN
 1000 C=0;IF PX(X,Y-1)C=C+1
 1010 IF PX(X-1,Y-1)C=C+1
 1020 IF PX(X-1,Y)C=C+1
 1030 IF PX(X-1,Y+1)C=C+1
 1040 IF PX(X,Y+1)C=C+1
 1050 IF PX(X+1,Y+1)C=C+1
 1060 IF PX(X+1,Y)C=C+1
 1070 IF PX(X+1,Y-1)C=C+1
 1080 RETURN
 2000 NT=0;CX=-11;CY=-16;PRINT #1,P," ";NT=3
 2010 IF P=0CY=-24;PRINT "COLONY";PRINT "TERMINATED";STOP
 2015 RETURN
```

FIRST CLASS